


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide



[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used local cache and message and version and copy

Found 89,702 of 198,310

Sort results by

☒ [Save results to a Binder](#)
[Try an Advanced Search](#)

Display results

☒ [Search Tips](#)
[Try this search in The ACM Guide](#)
☐ Open results in a new window

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

 Relevance scale ☐ ☐ ☐ ☐ ☐

### 1 [4.2BSD and 4.3BSD as examples of the UNIX system](#)


 John S. Quarterman; Abraham Silberschatz, James L. Peterson  
 December 1985 **ACM Computing Surveys (CSUR)**, Volume 17 Issue 4

Publisher: ACM Press

 Full text available: ☒ pdf(4.07 MB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This paper presents an in-depth examination of the 4.2 Berkeley Software Distribution, Virtual VAX-11 Version (4.2BSD), which is a version of the UNIX Time-Sharing System. There are notes throughout on 4.3BSD, the forthcoming system from the University of California at Berkeley. We trace the historical development of the UNIX system from its conception in 1969 until today, and describe the design principles that have guided this development. We then present the internal data structures and ...

### 2 [pHuid: the design of a parallel functional language implementation on workstations](#)


 Cormac Flanagan, Rishiyur S. Nikhil  
 June 1996 **ACM SIGPLAN Notices , Proceedings of the first ACM SIGPLAN international conference on Functional programming ICFP '96**, Volume 31 Issue 6

Publisher: ACM Press

 Full text available: ☒ pdf(1.20 MB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes the distributed memory implementation of a shared memory parallel functional language. The language is Id, an implicitly parallel, mostly functional language that is currently evolving into a dialect of Haskell. The target is a distributed memory machine, because we expect these to be the most widely available parallel platforms in the future. The difficult problem is to bridge the gap between the shared memory language model and the distributed memory machine model. The lan ...

**Keywords:** data flow, garbage collection and run-time systems, parallel and distributed implementations

### 3 [A caching file system for a programmer's workstation](#), principles that have


 Michael D. Schroeder, David K. Gifford, Roger M. Needham  
 December 1985 **ACM SIGOPS Operating Systems Review , Proceedings of the tenth ACM symposium on Operating systems principles SOSP '85**, Volume 19 Issue 5

10/726,295

**Publisher:** ACM Press

Full text available:  pdf(768.75 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

#### 4 A coherent distributed file cache with directory write-behind



Timothy Mann, Andrew Birrell, Andy Hisgen, Charles Jerian, Garret Swart

May 1994 **ACM Transactions on Computer Systems (TOCS)**, Volume 12 Issue 2

**Publisher:** ACM Press

Full text available:  pdf(3.21 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Extensive caching is a key feature of the Echo distributed file system. Echo client machines maintain coherent caches of file and directory data and properties, with write-behind (delayed write-back) of all cached information. Echo specifies ordering constraints on this write-behind, enabling applications to store and maintain consistent data structures in the file system even when crashes or network faults prevent some writes from being completed. In this paper we describe ...

**Keywords:** coherence, file caching, write-behind

#### 5 A comparison of message passing and shared memory architectures for data parallel programs



A. C. Klaiber, H. M. Levy

April 1994 **ACM SIGARCH Computer Architecture News , Proceedings of the 21ST annual international symposium on Computer architecture ISCA '94**, Volume 22 Issue 2

**Publisher:** IEEE Computer Society Press, ACM Press

Full text available:  pdf(1.26 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Shared memory and message passing are two opposing communication models for parallel multicomputer architectures. Comparing such architectures has been difficult, because applications must be hand-crafted for each architecture, often resulting in radically different sources for comparison. While it is clear that shared memory machines are currently easier to program, in the future, programs will be written in high-level languages and compiled to the specific parallel target, thus eliminating thi ...

#### 6 A comparison of MPI, SHMEM and cache-coherent shared address space programming models on the SGI Origin2000



Hongzhang Shan, Jaswinder Pal Singh

May 1999 **Proceedings of the 13th international conference on Supercomputing ICS '99**

**Publisher:** ACM Press


Full text available:  pdf(2.30 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

#### 7 A comparison of three programming models for adaptive applications on the Origin2000

Hongzhang Shan, Jaswinder P. Singh, Leonid Oliker, Rupak Biswas

November 2000 **Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM) Supercomputing '00**

**Publisher:** IEEE Computer Society

Full text available:  pdf(239.30 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)



Publisher Site

Adaptive applications have computational workloads and communication patterns which change unpredictably at runtime, requiring load balancing to achieve scalable performance on parallel machines. Efficient parallel implementation of such adaptive application is therefore a challenging task. In this paper, we compare the performance of and the programming effort required for two major classes of adaptive applications under three leading parallel programming models on an SGI Origin 2000 syste ...

# 8 A cost-comparison approach for adaptive distributed shared memory



Jai-Hoon Kim, Nitin H. Vaidya

January 1996 **Proceedings of the 10th international conference on Supercomputing ICS '96**

Publisher: ACM Press

Full text available: pdf(976.97 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

# 9 A framework for delivering multicast message in networks with mobile hosts

Arup Acharya, B. R. Badrinath

October 1996 **Mobile Networks and Applications**, Volume 1 Issue 2

Publisher: Kluwer Academic Publishers

Full text available: pdf(438.33 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

To accommodate mobile hosts (MHs) within existing data networks, the static network is augmented with "mobile support stations" (MSSs) that communicate directly with MHs, usually via wireless links. Connectivity of the overall network changes dynamically as MHs connect to the static network from different "locations" (MSSs) at different times. Compared to their desktop counterparts, mobile hosts face a new set of constraints namely, low bandwidth of the wireless links on an SGI Origin 2000 syste

# 10 A high performance cluster JVM presenting a pure single system image



Y. Aridor, M. Factor, A. Teperman, T. Eilam, A. Schuster

June 2000 **Proceedings of the ACM 2000 conference on Java Grande JAVA '00**

Publisher: ACM Press

Full text available: pdf(916.33 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

# 11 A high-level abstraction of shared accesses



Peter J. Keleher

February 2000 **ACM Transactions on Computer Systems (TOCS)**, Volume 18 Issue 1

Publisher: ACM Press

Full text available: pdf(183.57 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [review](#)

We describe the design and use of the tape mechanism, a new high-level abstraction of accesses to shared data for software DSMs. Tapes consolidate and generalize a number of recent protocol optimizations, including update-based locks and recorded-replay barriers. Tapes are usually created by "recording" shared accesses. The resulting recordings can be used to anticipate future accesses by tailoring data movement to application semantics. Tapes-based mechanisms a ...

**Keywords:** DSM, programming libraries, shared memory, update protocols

12 A message driven OR-parallel machine

S. A. Delgado-Rannauro, T. J. Reynolds

April 1989 **ACM SIGARCH Computer Architecture News , Proceedings of the third international conference on Architectural support for programming languages and operating systems ASPLOS-III**, Volume 17 Issue 2

Publisher: ACM Press

Full text available: pdf(1.29 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

A message driven architecture for the execution of OR-parallel logic languages is proposed. The computational model is based on well known compilation techniques for Logic Languages. We present first the multiple binding mechanism for the OR-parallel Prolog architecture and the corresponding OR-parallel abstract machine is described. A scheduling algorithm which does not rely upon the availability of global data structures to direct the search for work is discussed. The message driven proce ...

13 A Performance Evaluation of the Convex SPP-1000 Scalable Shared Memory Parallel Computer

Thomas Sterling, Daniel Savaresse, Peter MacNeice, Kevin Olson, Clark Mobarry, Bruce Fryxell, Phillip Merkey

December 1995 **Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM) - Volume 00 Supercomputing '95**

Publisher: ACM Press

Full text available: pdf(457.11 KB)

html(2.67 KB)

ps(780.23 KB)

Additional Information: [full citation](#), [references](#), [citing](#)14 A quantitative analysis of cache policies for scalable network file systems

Michael D. Dahlin, Clifford J. Mather, Randolph Y. Wang, Thomas E. Anderson, David A. Patterson

May 1994 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1994 ACM SIGMETRICS conference on Measurement and modeling of computer systems SIGMETRICS '94**, Volume 22 Issue 1

Publisher: ACM Press

Full text available: pdf(1.42 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citing](#), [index terms](#)

Current network file system protocols rely heavily on a central server to coordinate file activity among client workstations. This central server can become a bottleneck that limits scalability for environments with large numbers of clients. In central server systems such as NFS and AFS, all client writes, cache misses, and coherence messages are handled by the server. To keep up with this workload, expensive server machines are needed, configured with high-performance CPUs, memory systems, ...

15 A quantitative comparison of parallel computation models

Ben H. H. Juurlink, Harry A. G. Wijshoff

August 1998 **ACM Transactions on Computer Systems (TOCS)**, Volume 16 Issue 3

Publisher: ACM Press

Full text available: pdf(1.06 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citing](#), [index terms](#)

In recent years, a large number of parallel computation models have been proposed to replace the PRAM as the parallel computation model presented to the algorithm designer. Although mostly the theoretical justifications for these models are sound, and many algorithmic results were obtained through these models, little experimentation has been conducted to validate the effectiveness of these models for developing cost-effective

algorithms and applications on existing hardware platforms. In ...

**Keywords:** parallel computation models, performance evaluation

# 16 A quorum-consensus replication method for abstract data types



Maurice Herlihy

February 1986 **ACM Transactions on Computer Systems (TOCS)**, Volume 4 Issue 1

**Publisher:** ACM Press

Full text available: pdf(1.66 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Replication can enhance the availability of data in distributed systems. This paper introduces a new method for managing replicated data. Unlike many methods that support replication only for uninterpreted files, this method systematically exploits type-specific properties of objects such as sets, queues, or directories to provide more effective replication. Each operation requires the cooperation of a certain number of sites for its successful completion. A quorum for an operation is any s ...

# 17 A reliable multicast webcast protocol for multimedia collaboration and caching



L. Kristin Wright, Steven McCanne, Jay Lepreau

October 2000 **Proceedings of the eighth ACM international conference on Multimedia MULTIMEDIA '00**

**Publisher:** ACM Press

Full text available: pdf(1.10 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Large-scale, multi-point, multimedia conferencing applications designed to facilitate long-distance collaboration are enjoying growing popularity. Usually composed of real-time audio, video and shared-drawing applications, these collaborative environments help render the geographical location of collaborators irrelevant. To complement these existing collaborative applications, it would be useful to have the ability to distribute documents synchronously over the World Wide Web (WWW). One model ...

# 18 A scalable Web cache consistency architecture



Haobo Yu, Lee Breslau, Scott Shenker

August 1999 **ACM SIGCOMM Computer Communication Review, Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication SIGCOMM '99**, Volume 29 Issue 4.

**Publisher:** ACM Press

Full text available: pdf(1.71 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The rapid increase in web usage has led to dramatically increased loads on the network infrastructure and on individual web servers. To ameliorate these mounting burdens, there has been much recent interest in web caching architectures and algorithms. Web caching reduces network load, server load, and the latency of responses. However, web caching has the disadvantage that the pages returned to clients by caches may be *stale*, in that they may not be consistent with the version currently o ...

# 19 A structural view of the Cedar programming environment



Daniel C. Swinehart, Polle T. Zellweger, Richard J. Beach, Robert B. Hagmann

August 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 8 Issue 4

**Publisher:** ACM Press

Full text available: pdf(6.32 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents an overview of the Cedar programming environment, focusing on its

overall structure—that is, the major components of Cedar and the way they are organized. Cedar supports the development of programs written in a single programming language, also called Cedar. Its primary purpose is to increase the productivity of programmers whose activities include experimental programming and the development of prototype software systems for a high-performance personal computer. T ...

## 20 [A survey of commercial parallel processors](#)



Edward Gehringer, Janne Abullarade, Michael H. Gulyan

September 1988 **ACM SIGARCH Computer Architecture News**, Volume 16 Issue 4

**Publisher:** ACM Press

Full text available: pdf(2.96 MB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

This paper compares eight commercial parallel processors along several dimensions. The processors include four shared-bus multiprocessors (the Encore Multimax, the Sequent Balance system, the Alliant FX series, and the ELXSI System 6400) and four network multiprocessors (the BBN Butterfly, the NCUBE, the Intel IPSC/2, and the FPS T Series). The paper contrasts the computers from the standpoint of interconnection structures, memory configurations, and interprocessor communication: Also, the share ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)

## Dialing DataStar

[options](#)[logout](#)[feedback](#)[help](#)[databases](#)[easy search](#)

## Advanced Search:

Inspec - 1898 to date (INZZ)

[limit](#)

Search history:

No.	Database	Search term	Info added since	Results	
CP		[Clipboard]		0	-
1	INZZ	updating NEAR local ADJ cach\$	unrestricted	1	<a href="#">show titles</a>
2	INZZ	local ADJ cache	unrestricted	121	<a href="#">show titles</a>
3	INZZ	2 AND messag\$	unrestricted	11	<a href="#">show titles</a>



[hide](#) | [delete all search steps...](#) | [delete individual search steps...](#)Enter your search term(s): [Search tips](#) ☐ Thesaurus mapping whole document Information added since: or:  none [search](#)

(YYYYMMDD)

Inspec - 1898 to date (INZZ)

☐ Documents with images

Select special search terms from the following list(s):

- ☒ Publication year 1950-
- ☒ Publication year 1898-1949
- ☒ Inspec thesaurus - browse headings 
- ☒ Inspec thesaurus - enter a term 
- ☒ Classification codes A: Physics, 0-1
- ☒ Classification codes A: Physics, 2-3
- ☒ Classification codes A: Physics, 4-5
- ☒ Classification codes A: Physics, 6
- ☒ Classification codes A: Physics, 7
- ☒ Classification codes A: Physics, 8
- Classification codes A: Physics, 9

101726, 295

# Dial DataStar

options

logout

feedback

help

databases

search  
page

titles

## Document

Select the documents you wish to save or order by clicking the box next to the document, or click the link above the document to order directly.

save

locally as: PDF document



search strategy: do not include the search strategy



order

copy to  
Clipboard
☒ Select All

- 1 [Efficient cooperative caching in ad hoc networks.](#)
- 2 [Cooperative cache management in mobile ad hoc networks.](#)
- 3 [Cache updates in a peer-to-peer network of mobile agents.](#)
- 4 [Performance analysis of distributed search in open agent systems.](#)
- 5 [The costs of using JXTA.](#)
- 6 [The impact of caching in a loosely-coupled clustered software DSM system.](#)
- 7 [Shared memory versus message passing for iterative solution of sparse, irregular](#)
- 8 [A predicate-based caching scheme for client-server database architect](#)
- 9 [Dynamic self-invalidation: reducing coherence overhead in shared- memory mu](#)
- 10 [Implications of hierarchical N-body methods for multiprocessor architect](#)
- 11 [A predicate-based caching scheme for client-server database architect](#)

☒ document 1 of 11 [Order Document](#)

Inspec - 1898 to date (INZZ)

### Accession number & update

0009076035 20070101.

### Title

Efficient cooperative caching in ad hoc networks.

### Conference information

2006 1st International Conference on Communication System Software and Middleware, New Delhi, India, 8-12 Jan. 2006.

### Source

2006 1st International Conference on Communication System Software and Middleware (IEEE Cat. No. 06EX1230C), 2006, p. 8 pp., 26 refs, pp. CD-ROM, ISBN: 0-7803-9574-3.  
Publisher: IEEE, Piscataway, NJ, USA.

### Author(s)

Chand-N, Joshi-R-C, Misra-M.

### Author affiliation

Chand, N., Joshi, R.C., Misra, M., Dept. of Electron. & Comput. Eng., Indian Inst. of Technol., Roorkee, India.

### Abstract

Caching of frequently accessed data in multi-hop ad hoc environment is a potential technique that can improve the data access performance and availability. Cooperative caching, which allows the sharing and coordination of cached data among clients, can further explore the potential of the caching techniques. In this paper, we propose a novel scheme, called zone cooperative (ZC) for caching in mobile ad hoc networks. In ZC scheme, one-hop neighbors of a mobile client form a cooperative cache zone since the cost for communication with them is low both in terms of energy consumption and



**message** exchange. As a part of **cache** management, **cache** admission control and VALUE based replacement policy are developed to improve the data accessibility and reduce the **local cache** miss ratio. An analytical study of ZC based on data popularity, node density and transmission range is also performed. Simulation experiments show that the ZC caching mechanism achieves significant improvements in **cache** hit ratio and average query latency in comparison with other caching strategies.

**Descriptors**

 AD-HOC-NETWORKS;  CACHE-STORAGE;  INFORMATION-RETRIEVAL;  MOBILE-RADIO;  
 TELECOMMUNICATION-CONGESTION-CONTROL;  TELECOMMUNICATION-NETWORK-  
MANAGEMENT;  TELECOMMUNICATION-NETWORK-RELIABILITY.

**Classification codes**

B6250F Mobile-radio-systems\*;  
B1265D Memory-circuits;  
B6210C Network-management.

**Keywords**

zone-cooperative-caching; multihop-ad-hoc-environment; data-access-performance; data-availability;  
mobile-ad-hoc-network; MANET; energy-consumption; **message-exchange**; **cache-management**;  
**cache-admission-** control; VALUE-based-replacement-policy.

**Treatment codes**

P Practical;  
X Experimental.

**Language**

English.

**Publication type**

Conference-paper.

**Availability**

CCCC: 0-7803-9574-3/06/\$20.00.

**Publication year**

2006.

**Publication date**

20060000.

**Edition**

2006037.

**Copyright statement**

Copyright 2006 The Institution of Engineering and Technology.

(c) 2007 The Institution of Engineering and Technology

---

☒ **document 2 of 11** Order Document

**Inspec - 1898 to date (INZZ)**

**Accession number & update**

0009072920 20070101.

**Title**

Cooperative **cache** management in mobile ad hoc networks.

**Conference information**

IEE Mobility Conference 2005. The Second International Conference on Mobile Technology, Applications and Systems, Guangzhou, China, 15-17 Nov. 2005.

**Source**

IEE Mobility Conference 2005. The Second International Conference on Mobile Technology, Applications and Systems, 2005, p. 7 pp., 23 refs, pp. CD-ROM, ISBN: 981-05-4571-1.  
Publisher: IEE, Stevenage, UK.

**Author(s)**

Chand-N, Joshi-R-C, Misra-M.

**Author affiliation**

Chand, N.; Joshi, R.C.; Misra, M., Indian Inst. of Technol., Roorkee, India.

**Abstract**

Caching of frequently accessed data in ad hoc networks is a potential technique that can improve the data access performance and availability. Cooperative caching, which allows the sharing and coordination of cached data among clients, can further explore the potential of the caching techniques. In this paper, we propose a novel scheme, called zone cooperative (ZC) for caching in mobile ad hoc networks. In ZC scheme, one-hop neighbors of a mobile client form a cooperative **cache** zone since the cost for communication with them is low both in terms of energy consumption and **message** exchange. For a data miss in the **local cache**, each client first searches the data in its zone before forwarding the request to the next client that lies along routing path towards server. As a part of **cache** management, **cache** admission control and replacement policy are developed to improve the data accessibility and reduce the **local cache** miss ratio. Simulation experiments show that the ZC caching mechanism achieves significant improvements in terms of **cache** hit ratio and average query latency in comparison with other caching strategies.

**Descriptors**

 **AD-HOC-NETWORKS;**  **CACHE-STORAGE;**  **MOBILITY-MANAGEMENT-MOBILE-RADIO;**  
 **TELECOMMUNICATION-CONGESTION-CONTROL;**  **TELECOMMUNICATION-NETWORK-ROUTING.**

**Classification codes**

B6250F Mobile-radio-systems\*;  
B6210C Network-management;  
B6150P Communication-network-design-planning-and-routing;  
C3370H Control-applications-in-radio-and-radar\*;

**Keywords**

**cooperative-cache-management;** mobile-ad-hoc-networks; zone-cooperative; mobile-client;  
**message-exchange;** routing-path; **cache-** admission-control; replacement-policy; average-query-latency.

**Treatment codes**

P Practical.

**Language**

English.

**Publication type**

Conference-paper.

**Publication year**

2005.

**Publication date**

20050000.

**Edition**

2006037.

**Copyright statement**

Copyright 2006 The Institution of Engineering and Technology.

(c) 2007 The Institution of Engineering and Technology

☒ **document 3 of 11** [Order Document](#)

**Inspec - 1898 to date (INZZ)**

**Accession number & update**

0008272543 20070101.

**Title**

**Cache updates in a peer-to-peer network of mobile agents.**

**Conference information**

Proceedings. Fourth International Conference on Peer-to-Peer Computing, Zurich, Switzerland, 25-27 Aug. 2004.

Sponsor(s): IEEE; Swiss Federal Inst. of Technol. (ETH Zurich); Linkopings universitet; Credit Suisse Group; IBM Res; Sun Microsystems Lab.

**Source**

Proceedings. Fourth International Conference on Peer-to-Peer Computing, 2004, p. 10-17, 16 refs, pp.

x+286, ISBN: 0-7695-2156-8.

Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA.

**Author(s)**

Leontiadis-E, Dimakopoulos-V-V, Pitoura-E.

Editor(s): Caronni-G, Weiler-N, Shahmehri-N.

**Author affiliation**

Leontiadis, E., Dimakopoulos, V.V., Pitoura, E., Dept. of Comput. Sci., Ioannina Univ., Greece.

**Abstract**

In open multi-agent systems, agents need resources provided by other agents but they are not aware of which agents provide particular resources. We consider a peer-to-peer approach, in which each agent maintains a **local cache** with information about  $k$  resources, that is for each of the  $k$  resources, an agent that provides it. However, when an agent or a resource moves, **cache** entries become obsolete. We propose a suite of **cache** update policies that combine pull-based invalidation that is initiated by the agent that maintains the **cache** with push-based invalidation that is initiated by the agent that moves. We study and compare variations of oblivious flooding-based push/pull along with an informed push approach where each agent maintains a list of the agents that have it cached. Our experimental results indicate that a novel variation of flooding for push where a moving agent propagates its new location to agents in its old neighborhood achieves good **cache** consistency with a small **message** overhead. The proposed policies are suitable for any peer-to-peer system where peers **cache** information about other peers and this information becomes obsolete.

**Descriptors**

 CACHE-STORAGE;  MOBILE-AGENTS;  MULTI-AGENT-SYSTEMS;  PEER-TO-PEER-COMPUTING.

**Classification codes**

C6170 Expert-systems-and-other-AI-software-and-techniques\*;

C6150N Distributed-systems-software;

C6120 File-organisation.

**Keywords**

peer-to-peer-network; mobile-agents; open-multiagent-systems; **local- cache**; **cache-update-policies**; pull-based-invalidation; push-based- invalidation; flooding-based-push; flooding-based-pull; informed-push-approach; **cache-consistency**; information-caching.

**Treatment codes**

P Practical.

**Language**

English.

**Publication type**

Conference-paper.

**Availability**

CCCC: 0 7695 2156 8/2004/\$20.00.

**Publication year**

2004.

**Publication date**

20040000.

**Edition**

2005005.

**Copyright statement**

Copyright 2005 IEE.

(c) 2007 The Institution of Engineering and Technology

☒ **document 4 of 11** Order Document

**Inspec - 1898 to date (INZZ)**

**Accession number & update**

0007891278 20070101.

**Title**

Performance analysis of distributed search in open agent systems.

**Conference information**

International Parallel and Distributed Processing Symposium (IPDPS 2003), Nice, France, 22-26 April 2003.

Sponsor(s): IEEE Comput. Soc Tech. Committee on Parallel Process; IEEE Comput. Soc. Tech. Committee on Comput. Archit; IEEE Comput. Soc. Tech. Committee on Distrib. Process; ACM SIGARCH.

**Source**

Proceedings International Parallel and Distributed Processing Symposium, 2003, p. 8 pp., 13 refs, pp. CD-ROM, ISBN: 0-7695-1926-1.

Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA.

**Author(s)**

Dimakopoulos-V-V, Pitoura-E.

**Author affiliation**

Dimakopoulos, V.V., Pitoura, E., Dept. of Comput. Sci., Ioannina Univ., Greece.

**Abstract**

In open multi-agent systems agents need resources provided by other agents but they are not aware of which agents provide the particular resources. Most solutions to this problem are based on a central directory that maintains a mapping between agents and resources. However, such solutions do not scale well since the central directory becomes a bottleneck in terms of both performance and reliability. In this paper, we introduce a different approach: each agent maintains a limited size **local cache** in which it keeps information about  $k$  different resources, that is, for each of  $k$  resources, it stores the contact information of one agent that provides it. This creates a directed network of caches. We address the following fundamental problem: how can an agent that needs a particular resource find an agent that provides it by navigating through this network of caches? We propose and analytically compare the performance of three different algorithms for this problem, flooding, teeming and random paths, in terms of three performance measures: the probability to locate the resource, the number of steps and the number of **messages** to do so. Our analysis is also applicable to distributed search in unstructured peer-to-peer networks.

**Descriptors**

CACHE-STORAGE; COMPUTER-NETWORKS; DISTRIBUTED-ALGORITHMS; MULTI-AGENT-SYSTEMS; PERFORMANCE-EVALUATION; PROBABILITY; SEARCH-PROBLEMS.

**Classification codes**

C6170 Expert-systems-and-other-AI-software-and-techniques\*;

C5670 Network-performance;

C6120 File-organisation;

C6150N Distributed-systems-software.

**Keywords**

performance-analysis; distributed-search; open-agent-systems; multi-agent-systems; **limited-size-local-cache**; directed-network; flooding; resource-location-probability; step-number; **message-number**; unstructured-peer-to-peer-networks; teeming; random-paths; performance-measures.

**Treatment codes**

P Practical.

**Language**

English.

**Publication type**

Conference-paper.

**Publication year**

2003.

**Publication date**

20030000.

**Edition**

2004010.

**Copyright statement**

Copyright 2004 IEE.

(c) 2007 The Institution of Engineering and Technology

☒ **document 5 of 11** Order Document

**Inspec - 1898 to date (INZZ)**

**Accession number & update**

0007861634 20070101.

**Title**

The costs of using JXTA.

**Conference information**

Proceedings Third International Conference on Peer-to-Peer Computing (P2P2003), Linkoping, Sweden, 1-3 Sept. 2003.

Sponsor(s): IEEE Sweden; Linkopings Univ; IISLAB (Intelligent Inf. Syst. Lab.).

**Source**

Proceedings Third International Conference on Peer-to-Peer Computing (P2P2003), 2003, p. 160-7, 15 refs, pp. x+222, ISBN: 0-7695-2023-5.

Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA.

**Author(s)**

Halepovic-E, Deters-R.

Editor(s): Shahmehri-N, Graham-R-L, Carroni-G.

**Author affiliation**

Halepovic, E., Deters, R., Dept. of Comput. Sci., Saskatchewan Univ., Saskatoon, Sask., Canada.

**Abstract**

Project JXTA is an open-source effort to specify the standard protocols for peer-to-peer communication and collaboration. We propose a JXTA performance model and present results obtained by benchmarking the JXTA 1.0 reference implementation in Java. We focus on the performance evaluation of typical peer operations and consequences for the peer network, the user and the developer. The important trade-off between peer startup latency and the maintenance of the **local cache** is shown and discussed. The throughput limits of pipes, the core JXTA communication concept, are also measured in a LAN environment for smooth and bursty traffic. The results indicate that the limiting factor for reliable throughput is the number of **messages** rather than size in bytes, as well as that small JXTA **messages** carry an excessive overhead of control data. Important performance issues and trade-offs are identified and explored, as a basis for the formulation of guidelines for system designers and simulation-based research of JXTA networks.

**Descriptors**

CACHE-STORAGE; JAVA; LOCAL-AREA-NETWORKS; MESSAGE-PASSING; OPEN-SYSTEMS; PERFORMANCE-EVALUATION; PROTOCOLS; SYSTEMS-ANALYSIS; TELECOMMUNICATION-NETWORK-RELIABILITY; TELECOMMUNICATION-TRAFFIC.

**Classification codes**

C5670 Network-performance\*;

C6150N Distributed-systems-software;

C5640 Protocols;

C5620L Local-area-networks.

**Keywords**

JXTA-network; standard-protocol; peer-to-peer-communication; peer-to-peer-collaboration; JXTA-performance-model; Java; peer-startup-latency; **local-cache**; LAN-environment; **JXTA-message**; simulation-based-research.

**Treatment codes**

P Practical.

**Language**

English.

**Publication type**

Conference-paper.

**Availability**

CCCC: 0 7695 2023 5/2003/\$17.00.

**Publication year**

2003.

**Publication date**

20030000.

**Edition**

2004006.

**Copyright statement**

Copyright 2004 IEE.

(c) 2007 The Institution of Engineering and Technology

☒ **document 6 of 11** Order Document

**Inspec - 1898 to date (INZZ)**

**Accession number & update**

0006805970 20070101.

**Title**

The impact of caching in a loosely-coupled clustered software DSM system.

**Conference information**

Proceedings IEEE International Conference on Cluster Computing. CLUSTER 2000, Chemnitz, Germany, 28 Nov.-1 Dec. 2000.

**Source**

Proceedings IEEE International Conference on Cluster Computing. CLUSTER 2000, 2000, p. 27-34, 11 refs, pp. xxvii+406, ISBN: 0-7695-0896-0.

Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA.

**Author(s)**

Arantes-L, Sens-P, Folliot-B.



**Author affiliation**

Arantes, L., Sens, P., Folliot, B., LIP6 Lab., Paris VI Univ., France.

**Abstract**

As interconnected **local-area** workstation networks are widely available, the idea of offering a software distributed shared memory (SDSM) layer across them is quite an attractive alternative for compute-intensive applications. However, the higher cost of sending a **message** over an inter-cluster link than over an intraccluster can limit applications' performance on a multicluster SDSM system. In this paper, we present the extensions that we have added to TreadMarks SDSM in order to adapt it to a loosely-coupled cluster-based platform. We have implemented a logical per-cluster **cache** in order to exploit cluster locality. By accessing its **local cache**, a processor can share data previously requested by another processor of its cluster, thereby hiding the cost of inter-cluster communication.

**Descriptors**

 CACHE-STORAGE;  DISTRIBUTED-SHARED-MEMORY-SYSTEMS;  LAN-INTERCONNECTION;  WORKSTATION-CLUSTERS.

**Classification codes**

B6210L Computer-communications\*;

C5620L Local-area-networks\*;

C5440 Multiprocessing-systems;

C6120 File-organisation;

C6150N Distributed-systems-software.

**Keywords**

caching; loosely-coupled-clustered-software-distributed-shared-memory-; **interconnected-local-area-workstation-networks**; compute-intensive-applications; TreadMarks-SDSM; **logical-per-cluster-cache**; inter-cluster-communication.

**Treatment codes**

A Application;

P Practical.

**Language**

English.

**Publication type**

Conference-paper.

**Availability**

CCCC: 0 7695 0896 0/2000/\$10.00.

**Digital object identifier**

10.1109/CLUSTER.2000.888989.

**Publication year**

2000.

**Publication date**

20000000.

**Edition**

2001001.

**Copyright statement**

Copyright 2001 IEE.

(c) 2007 The Institution of Engineering and Technology

Full text available at

Electronic  
Journals  
Service

USPTO Full Text Retrieval Options

☒ **document 7 of 11** Order Document**Inspec - 1898 to date (INZZ)****Accession number & update**

0006344294 20070101.

**Title**Shared memory versus **message** passing for iterative solution of sparse, irregular problems.**Source**Parallel Processing Letters, {Parallel-Process-Lett-Singapore}, March 1999, vol. 9, no. 1, p. 159-70, 15 refs, CODEN: PPLTEE, ISSN: 0129-6264.  
Publisher: World Scientific, Singapore.**Author(s)**

Chong-F-T, Agarwal-A.

**Author affiliation**

Chong, F.T., Dept. of Comput. Sci., California Univ., Davis, CA, USA.

**Abstract**

The benefits of hardware support for shared memory versus those for **message** passing are difficult to evaluate without an in-depth study of real applications on a common platform. We evaluate the communication mechanisms of the MIT Alewife machine, a multiprocessor which provides integrated **cache-coherent** shared memory, **message** passing, and DMA. We perform this evaluation with 'best-effort' implementations which solve several sparse, irregular benchmark problems with a preconditioned conjugate gradient sparse matrix solver (ICCG). We find that machines with fast global memory operations do not need **message** passing or bulk transfer to support our irregular problems. This is primarily due to three reasons. First, a 5-to-1 ratio between global and **local cache** misses makes memory copies in bulk communication expensive relative to communication via shared memory. Second, although **message** passing has synchronization semantics superior to shared memory for data-driven computation, efficient shared memory can overcome this handicap by using global read-modify-writes to change from the traditional owner-computes model to a producer-computes model. Third, bulk transfers can result in high processor idle times in irregular applications.

**Descriptors**

MESSAGE-PASSING; PERFORMANCE-EVALUATION; SHARED-MEMORY-SYSTEMS;  
 SPARSE-MATRICES.

**Classification codes**

C5440 Multiprocessing-systems\*;  
C5220P Parallel-architecture;  
C5470 Performance-evaluation-and-testing;  
C4140 Linear-algebra-numerical-analysis.

**Keywords**

shared-memory; **message-passing**; MIT-Alewife-machine; communication-mechanisms;  
multiprocessor; **integrated-cache-coherent-shared-memory**; DMA; sparse-matrix-solver.

**Treatment codes**

P Practical.

**Language**

English.

**Publication type**

Journal-paper.

**Availability**

SICI: 0129-6264(199903)9:1L:159:SMVM; 1-Z.

**Publication year**

1999.

**Publication date**

19990300.

**Edition**

1999035.

**Copyright statement**

Copyright 1999 IEE.

(c) 2007 The Institution of Engineering and Technology

Full text available at **USPTO Full Text Retrieval Options**

☒ **document 8 of 11** Order Document

**Inspec - 1898 to date (INZZ)**

**Accession number & update**

0005403647 20070101.

**Title**

A predicate-based caching scheme for client-server database architectures.

**Source**

VLDB Journal, {VLDB-J-Germany}, Jan. 1996, vol. 5, no. 1, p. 35-47, 34 refs, ISSN: 1066-8888.

Publisher: Springer-Verlag, Germany.

**Author(s)**

Keller-A-M, Basu-J.






**Author affiliation**

Keller, A.M., Dept. of Comput. Sci., Stanford Univ., CA.

**Abstract**

We propose a new client-side data-caching scheme for relational databases with a central server and multiple clients. Data are loaded into each client **cache** based on queries executed on the central database at the server. These queries are used to form predicates that describe the **cache** contents. A subsequent query at the client may be satisfied in its **local cache** if we can determine that the query result is entirely contained in the **cache**. This issue is called **cache completeness**. A separate issue, **cache currency**, deals with the effect on client caches of updates committed at the central database. We examine the various performance tradeoffs and optimization issues involved in addressing the questions of **cache** currency and completeness using predicate descriptions and suggest solutions that promote good dynamic behavior. Lower query-response times, reduced **message** traffic, higher server throughput, and better scalability are some of the expected benefits of our approach over commonly used relational server-side and object ID-based or page-based client-side caching.

**Descriptors**

 CACHE-STORAGE;  CLIENT-SERVER-SYSTEMS;  CONCURRENCY-CONTROL;  QUERY-PROCESSING;  RELATIONAL-DATABASES.

**Classification codes**

C6160D Relational-databases\*;

C6120 File-organisation;

C6160B Distributed-databases.

**Keywords**

client-side-data-caching-scheme; predicate-based-caching-scheme; client-server-database-architectures; relational-databases; central-server; multiple-clients; queries; central-database; **cache-contents**; **local-cache**; **cache-completeness**; **cache-currency**; updates; performance- tradeoffs; optimization; dynamic-behavior; query-response-times; **message-traffic**; server-throughput;



scalability.

**Treatment codes**

P Practical.

**Language**

English.

**Publication type**

Journal-paper.

**Availability**

SICI: 1066-8888(199601)5:1L:35:PBCS; 1-V.

**Publication year**

1996.

**Publication date**

19960100.

**Edition**

1996042.

**Copyright statement**

Copyright 1996 IEE.

(c) 2007 The Institution of Engineering and Technology

---

☒ **document 9 of 11** Order Document

**Inspec - 1898 to date (INZZ)**

**Accession number & update**

0005086781 20070101.

**Title**

Dynamic self-invalidation: reducing coherence overhead in shared-memory multiprocessors.

**Conference information**

Proceedings 22nd Annual International Symposium on Computer Architecture, Santa Margherita Ligure, Italy, 22-24 June 1995.

Sponsor(s): ACM SIGARCH; IEEE Comput. Soc., TCCA; Univ. Genoa.

**Source**

Proceedings 22nd Annual International Symposium on Computer Architecture (IEEE Cat. No.95CB35801), 1995, p. 48-59, 40 refs, pp. xiii+426, ISBN: 0-89791-698-0.

Publisher: ACM, New York, NY, USA.

**Author(s)**

Lebeck-A-R, Wood-D-A.

**Author affiliation**

Lebeck, A.R., Wood, D.A., Dept. of Comput. Sci., Wisconsin Univ., Madison, WI, USA.

**Abstract**

The paper introduces dynamic self-invalidation (DSI), a new technique for reducing **cache** coherence overhead in shared-memory multiprocessors. DSI eliminates invalidation **messages** by having a processor automatically invalidate its **local** copy of a **cache** block before a conflicting access by another processor. Eliminating invalidation overhead is particularly important under sequential consistency: where the latency of invalidating outstanding copies can increase a program's critical path. DSI is applicable to software, hardware, and hybrid coherence schemes. We evaluate DSI in the context of hardware directory-based write-invalidate coherence protocols. Our results show that DSI reduces execution time of a sequentially consistent full-map coherence protocol by as much as 41%. This is comparable to an implementation of weak consistency that uses a coalescing write-buffer to allow up to 16 outstanding requests for exclusive blocks. When used in conjunction with weak consistency DSI can exploit tear-off blocks-which eliminate both invalidation and acknowledgment **messages-for** a total reduction in **messages** of up to 26%.

**Descriptors**

 CACHE-STORAGE;  PROTOCOLS;  SHARED-MEMORY-SYSTEMS;  SOFTWARE-PERFORMANCE-EVALUATION.

**Classification codes**

C6150N Distributed-systems-software\*;  
C5640 Protocols;  
C6120 File-organisation.

**Keywords**

dynamic-self-invalidation; shared-memory-multiprocessors; **cache- coherence-overhead-reduction**; **invalidation-message-elimination**; **local- cache-block-copy**; conflicting-access; sequential-consistency; outstanding-copies; program-critical-path; hybrid-coherence-schemes; hardware; software; hardware-directory-based-write-invalidate-coherence-protocols; execution-time; sequentially-consistent-full-map-coherence-protocol; coalescing-write-buffer; outstanding-requests; exclusive-blocks; tear-off-blocks; **acknowledgment-message-elimination**.

**Treatment codes**

P Practical.

**Language**

English.

**Publication type**

Conference-paper.

**Availability**

CCCC: 0 89791 698 0/95/0006.\$3.50.

**Digital object identifier**

10.1109/ISCA.1995.524548.

**Publication year**

1995.

**Publication date**

19950000.

**Edition**

1995041.

**Copyright statement**

Copyright 1995 IEE.

(c) 2007 The Institution of Engineering and Technology

Full text available at



USPTO Full Text Retrieval Options

☒ **document 10 of 11** Order Document

**Inspec - 1898 to date (INZZ)**

**Accession number & update**

0005060184 20070101.

**Title**

Implications of hierarchical N-body methods for multiprocessor architectures.

**Source**

ACM Transactions on Computer Systems, {ACM-Trans-Comput-Syst-USA}, May 1995, vol. 13, no. 2, p. 141-202, 37 refs, CODEN: ACSYEC, ISSN: 0734-2071, USA.

**Author(s)**

Singh-J-P, Hennessy-J-L, Gupta-A.

**Author affiliation**





Singh, J.P., Hennessy, J.L., Gupta, A., Comput. Syst. Lab., Stanford Univ., CA, USA.

**Abstract**

To design effective large scale multiprocessors, designers need to understand the characteristics of the applications that will use the machines. One important class of applications is based on hierarchical N body methods. The article examines the key architectural implications of representative applications that use the two dominant hierarchical N body methods: the Barnes-Hut Method (J.E. Barnes, P. Hut, 1989) and the Fast Multipole Method. We first show that exploiting temporal locality on accesses to communicated data is critical to obtaining good performance on these applications and then argue that coherent caches on shared address space machines exploit this locality both automatically and very effectively. Next, we examine the implications of scaling the applications to run on larger machines. We

use scaling methods that reflect the concerns of the application scientist and find that this leads to different conclusions about how communication traffic and **local cache** and memory usage scale than scaling based only on data set size. In particular, we show that under the most realistic form of scaling, both the communication to computation ratio as well as the working set size grow slowly as larger problems are run on larger machines. Finally, we examine the effects of using the two dominant abstractions for interprocessor communication: a shared address space and explicit **message** passing between private address spaces.

**Descriptors**

 COMMUNICATION-COMPLEXITY;  MESSAGE-PASSING;  MULTIPROCESSING-SYSTEMS;  
 PARALLEL-ARCHITECTURES.

**Classification codes**

C5220P Parallel-architecture\*;  
C5440 Multiprocessing-systems;  
C6150N Distributed-systems-software;  
C4240C Computational-complexity;  
C4240P Parallel-programming-and-algorithm-theory.

**Keywords**

hierarchical-N-body-methods; multiprocessor-architectures; large-scale-multiprocessors; hierarchical-N-body-methods; architectural-implications; Barnes-Hut-Method; Fast-Multipole-Method; temporal-locality; coherent-caches; shared-address-space-machines; scaling-methods; communication-traffic; **local-cache**; memory-usage-scale; data-set-size; communication-to-computation-ratio; working-set-size; dominant-abstractions; interprocessor-communication; shared-address-space; **explicit-message-passing**; private-address-spaces.

**Treatment codes**

P Practical.

**Language**

English.

**Publication type**

Journal-paper.

**Availability**

CCCC: 0734-2071/95/0500-0141\$03.50.

**Publication year**

1995.

**Publication date**

19950500.

**Edition**

1995038.

**Copyright statement**

Copyright 1995 IEE.

(c) 2007 The Institution of Engineering and Technology

☒ **document 11 of 11** Order Document

**Inspec - 1898 to date (INZZ)**

**Accession number & update**

0004784874 20070101.

**Title**

A predicate-based caching scheme for client-server database architectures.

**Conference information**

Proceedings of 3rd International Conference on Parallel and Distributed Information Systems, Austin, TX, USA, 28-30 Sept. 1994.

Sponsor(s): IEEE Comput. Soc. Tech. Committee on Data Eng; ACM SIGMOD; Bellcore; US West.

**Source**

Proceedings of the Third International Conference on Parallel and Distributed Information Systems (Cat. No.94TH0668-4), 1994, p. 229-38, 22 refs, pp. xii+272, ISBN: 0-8186-6400-2.

Publisher: IEEE Comput. Soc. Press, Los Alamitos, CA, USA.

**Author(s)**

Keller-A-M, Basu-J.


**Author affiliation**

Keller, A.M., Dept. of Comput. Sci., Stanford Univ., Palo Alto, CA, USA.

**Abstract**

We propose a new client-side data caching scheme for relational databases with a central server and multiple clients. Data is loaded into a client **cache** based on queries, which are used to form predicates describing the **cache** contents. A subsequent query at the client may be satisfied in its **local cache** if we can determine that the query result is entirely contained in the **cache**. This issue is called **cache completeness**. On the other hand, **cache currency** deals with the effect of updates at the central database on the client caches. We examine various performance and optimization issues involved in addressing the questions of **cache** currency and completeness using predicate descriptions. Expected benefits of our approach over commonly used object 1D-based caching include lower query response times, reduced **message** traffic, higher server throughput, and better scalability.

**Descriptors**

 BUFFER-STORAGE;  DISTRIBUTED-DATABASES;  NETWORK-OPERATING-SYSTEMS;  
 RELATIONAL-DATABASES.

**Classification codes**

C6160B Distributed-databases\*;

C6160D Relational-databases;

C6150N Distributed-systems-software.

**Keywords**

predicate-based-caching-scheme; client-server-database-architectures; relational-databases; **cache-completeness**; query-response-times; **message-traffic**; server-throughput; scalability.

**Treatment codes**

P Practical.

**Language**

English.

**Publication type**

Conference-paper.

**Availability**

CCCC: 0 8186 6400 2/94/\$4.00.

**Digital object identifier**

10.1109/PDIS.1994.331711.

**Publication year**

1994.

**Publication date**

19940000.

**Edition**

1994040.

**Copyright statement**

Copyright 1994 IEE.

(c) 2007 The Institution of Engineering and Technology

save

locally as: PDF document



search strategy:

do not include the search strategy



order

copy to  
Clipboard

Top - News & FAQs - Dialog

© 2007 Dialog

# Dial · g DataStar

options

logout

feedback

help

databases

search  
page

titles

## Document

Select the documents you wish to save or order by clicking the box next to the document, or click the link above the document to order directly.

save

locally as: PDF document



search strategy: do not include the search strategy



order

copy to  
Clipboard

Full text available at **USPTO Full Text Retrieval Options**

☒ document 1 of 1 [Order Document](#)

**Inspec - 1898 to date (INZZ)**

### Accession number & update

0002374834 20070101.

### Title

Method for transmitting only document change data (for **local cache**- supported interactive terminals).

### Source

IBM Technical Disclosure Bulletin, {IBM-Tech-Disc-Bull-USA}, June 1984, vol. 27, no. 1B, p. 844-6, 0 refs, CODEN: IBMTAA, ISSN: 0018-8689, USA.

### Author(s)

Gladney-H-M.

### Author affiliation

Gladney, H.M., IBM Corp., Armonk, NY, USA.

### Abstract

The method steps include (a) locally **caching** documents and **updating** **local** logs and directories, (b) responsive to **updating** of the **local cache** documents through a counterpart terminal, comparing the form and content identities of the changed document with designated other **cache** directories, and (c) upon a match entering only the updates, upon a complete mismatch entering the entire document and updates, and upon a forms identity match entering only the content updates.

### Descriptors

**COMPUTER-NETWORKS;** **DATA-COMMUNICATION-SYSTEMS;** **INTERACTIVE-TERMINALS;** **PROTOCOLS.**

### Classification codes

**B6210L Computer-communications\*;**  
**C5620 Computer-networks-and-techniques\*.**

### Keywords

computer-networks; **local-cache-supported-terminals;** protocols; data-transmission-systems; document-change-data; interactive-terminals; **updating;** logs; directories; **local-cache-documents;** forms-identity- match.

### Treatment codes

P **Practical.**

### Language

English.

### Publication type

Journal-paper.

**Publication year**

1984.

**Publication date**

19840600.

**Edition**

1985004.

**Copyright statement**

Copyright 1985 IEE.

(c) 2007 The Institution of Engineering and Technology

save

locally as: PDF document



search strategy: do not include the search strategy



order

copy to  
Clipboard

Top - News &amp; FAQs - Dialog

© 2007 Dialog

Cheryl Lewis is logged in  
Logout
[Home + Recent Actions](#)   [Browse](#)   [Search](#)   [My Settings](#)   [Alerts](#)   [Help](#)

Quick Search   Title, abstract, keywords    Author  e.g.  
 ? search tips   Journal/book title    Volume    Issue    Page   
 results 1 - 18

## 18 Articles Found

(pub-date &gt; 1996 and TITLE-ABSTR-KEY(local cache)) and message

[Edit Search](#) | [Save Search](#) | [Save as Search Alert](#)      [Search With](#)

☒ = Full-text available   ☐ = Non-subscribed   ? What does this mean?

[Article List](#)   [Full Abstracts](#)
[display checked docs](#)   [e-mail articles](#)   [export citations](#)
Sort By:  [Go](#)

- ☒ 1. ☐ **Path conscious caching of B<sup>+</sup> tree indexes in a shared disks cluster • ARTICLE**  
*Journal of Parallel and Distributed Computing, Volume 67, Issue 3, March 2007, Pages 286-301*  
 Kyungoh Ohn and Haengrae Cho  
[Abstract](#) | [Abstract + References](#) | [PDF \(2258 K\)](#)

---

- ☒ 2. ☐ **An Accelerated IEEE 802.11 Handoff Process Based on the Dynamic Cluster Chain Method • ARTICLE**  
*Computer Communications, In Press, Corrected Proof, Available online 10 January 2007,*  
 Chung-Ming Huang and Jian-Wei Li  
[SummaryPlus](#) | [Full Text + Links](#) | [PDF \(706 K\)](#)

---

- ☒ 3. ☐ **Piggybacking related domain names to improve DNS performance • ARTICLE**  
*Computer Networks, Volume 50, Issue 11, 10 August 2006, Pages 1733-1748*  
 Hao Shang and Craig E. Wills  
[SummaryPlus](#) | [Full Text + Links](#) | [PDF \(596 K\)](#)

---

- ☒ 4. ☐ **A light-weight, collaborative temporary file system for clustered Web servers • ARTICLE**  
*Journal of Parallel and Distributed Computing, Volume 66, Issue 5, May 2006, Pages 750-762*  
 Jun Wang  
[Abstract](#) | [Abstract + References](#) | [PDF \(300 K\)](#)

---

- ☒ 5. ☐ **Locality in structured peer-to-peer networks • ARTICLE**  
*Journal of Parallel and Distributed Computing, Volume 66, Issue 2, February 2006, Pages 257-273*

10/726,295

Ronaldo A. Ferreira, Suresh Jagannathan and Ananth Grama  
 Abstract | Abstract + References | PDF (371 K)

---

6. ☐ **Load mitigation in cellular data networks by peer data sharing over WLAN channels • ARTICLE**

*Computer Networks, Volume 47, Issue 2, 4 February 2005, Pages 271-286*  
 Kang-Won Lee, Young-Bae Ko and Thyaga Nandagopal  
 SummaryPlus | Full Text + Links | PDF (426 K)

---

7. ☐ **The diffusion space of data diffusion architectures • ARTICLE**

*Parallel Computing, Volume 30, Issue 11, November 2004, Pages 1169-1193*  
 Jorge Buenabad-Chávez, Henk L. Muller, Paul W.A. Stallard and David H.D. Warren  
 SummaryPlus | Full Text + Links | PDF (295 K)

---

8. ☐ **Adaptive multiversion data broadcast organizations • ARTICLE**

*Information Systems, Volume 29, Issue 6, September 2004, Pages 509-528*  
 Oleg Shigiltchhoff, Panos K. Chrysanthis and Evaggelia Pitoura  
 SummaryPlus | Full Text + Links | PDF (660 K)

---

9. ☐ **On scalable and locality-aware web document sharing • ARTICLE**

*Journal of Parallel and Distributed Computing, Volume 63, Issue 10, October 2003, Pages 945-962*  
 Li Xiao, Xin Chen, Xiaodong Zhang and Yunhao Liu  
 SummaryPlus | Full Text + Links | PDF (265 K)

---

10. ☐ **Optimistic parallel simulation of a large-scale view storage system • ARTICLE**

*Future Generation Computer Systems, Volume 19, Issue 4, May 2003, Pages 479-492*  
 Garrett Yaun, Christopher D. Carothers, Sibel Adali and David Spooner  
 SummaryPlus | Full Text + Links | PDF (284 K)

---

11. ☐ **Using reordering technique for mobile transaction management in broadcast environments • ARTICLE**

*Data & Knowledge Engineering, Volume 45, Issue 1, April 2003, Pages 79-100*  
 SungSuk Kim, SangKeun Lee and Chong-Sun Hwang  
 SummaryPlus | Full Text + Links | PDF (544 K)

---

12. ☐ **Implementation and evaluation of directory hints in CC-NUMA multiprocessors • ARTICLE**

*Parallel Computing, Volume 28, Issue 1, January 2002, Pages 107-132*  
 Hung-Chang Hsiao and Chung-Ta King  
 SummaryPlus | Full Text + Links | PDF (188 K)

---

13. ☐ **WebCaL — a domain specific language for web caching • ARTICLE**

*Computer Communications, Volume 24, Issue 2, 1 February 2001, Pages 191-201*  
 S Gulwani, A Tarachandani, D Gupta, D Sanghi, L. P Barreto, G Muller, C Consel and Compose group  
 SummaryPlus | Full Text + Links | PDF (216 K)

---

**Toward Integration of Data Race Detection in DSM Systems • ARTICLE**



- ☒ 14. ☐

***Journal of Parallel and Distributed Computing, Volume 59, Issue 2, November 1999, Pages 180-203***

Ayal Itzkovitz, Assaf Schuster and Oren Zeev-Ben-Mordehai

[Abstract](#) | [Abstract + References](#) | [PDF \(385 K\)](#)

---

☒ 15. ☐

**Approximate bounds and expressions for the link utilization of shortest-path multicast network traffic • ARTICLE**

*Performance Evaluation, Volume 35, Issues 1-2, 1 March 1999, Pages 49-74*

Borislav H. [Reference to Simov], Srini B. [Reference to Tridandapani] and Michael S. [Reference to Borella]

[Abstract](#) | [PDF \(613 K\)](#)

---

☒ 16. ☐

**When to kill your siblings: cache mesh relation analysis • ARTICLE**

*Computer Networks and ISDN Systems, Volume 30, Issues 22-23, 25 November 1998, Pages 2105-2111*

Ingrid Melve

[SummaryPlus](#) | [Full Text + Links](#) | [PDF \(156 K\)](#) [STR-KEY\(local cache\)](#)

---

☒ 17. ☐

**Adaptive web caching: towards a new global caching architecture • ARTICLE**

*Computer Networks and ISDN Systems, Volume 30, Issues 22-23, 25 November 1998, Pages 2169-2177*

Scott Michel, Khoi Nguyen, Adam Rosenstein, Lixia Zhang, Sally Floyd and Van Jacobson

[SummaryPlus](#) | [Full Text + Links](#) | [PDF \(160 K\)](#)

---

☒ 18. ☐

**CFTP: a caching FTP server • ARTICLE**

*Computer Networks and ISDN Systems, Volume 30, Issues 22-23, 25 November 1998, Pages 2211-2222*

Mark Russell and Tim Hopkins

[SummaryPlus](#) | [Full Text + Links](#) | [PDF \(98 K\)](#)

## 18 Articles Found

(pub-date > 1996 and TITLE-ABSTR-KEY(local cache)) and message

[Edit Search](#) | [Save Search](#) | [Save as Search Alert](#)

results **1 - 18**

[Home + Recent Actions](#) | [Browse](#) | [Search](#) | [My Settings](#) | [Alerts](#) | [Help](#)



[About ScienceDirect](#) | [Contact Us](#) | [Terms & Conditions](#) | [Privacy Policy](#)

Copyright © 2007 Elsevier B.V. All rights reserved. ScienceDirect® is a registered trademark of Elsevier B.V.

ProQuest

[Return to the USPTO NPL Page](#) | [Help](#)

Basic

Advanced

Topics

Publications


 My Research  
0 marked items













Interface language:

English



Databases selected: Multiple databases...

**Results** – powered by ProQuest® Smart Search**Suggested Topics** [About](#)< Previous | [Next >](#)**Browse Suggested Publications** [About](#)< Previous | [Next >](#)[Cache](#)[Cache AND Computer science](#)[Cache AND Internet](#)[Cache AND Algorithms](#)[M2 Presswire; Coventry](#)[The New Yorker; New York](#)[IEEE Micro; Los Alamitos](#)5 documents found for: *local cache and message* » [Refine Search](#) | [Set Up Alert](#)  All sources  Trade Publications  Newspapers  Dissertations☐ Mark all  0 marked items: Email / Cite / Export Show only full textSort results by: **Most rec**

- 
- ☐ 1. **[Thin Client, Rich Data](#)**  
*Jon Udell. InfoWorld. San Mateo: Jun 14, 2004. Vol. 26, Iss. 24; p. 36 (1 page)*  
 [Text+Graphics](#)  [Full Text - PDF](#)  [Abstract](#)
- 
- ☐ 2. **[Hyper computing: a roadmap to the 21st century](#)**  
*The Hindu. Chennai: Jan 2, 1997. p. 1*  
 [Full text](#)  [Abstract](#)
- 
- ☐ 3. **[Parallel communication mechanisms for sparse, irregular applications](#)**  
*by Chong, Frederic T., Ph.D., Massachusetts Institute of Technology, 1997; AAT 0597970*  
 [Abstract](#)
- 
- ☐ 4. **[Verifying cache coherence protocols](#)**  
*McMillan, Kenneth. IEEE Spectrum. New York: Jun 1996. Vol. 33, Iss. 6; p. 67 (1 page)*  
 [Link to full text](#)  [Abstract](#)
- 
- ☐ 5. **[A novel and efficient distributed shared-memory algorithm for strict causal consistency](#)**  
*by Noseworthy, J. Russell, Ph.D., Rensselaer Polytechnic Institute, 1996, 228 pages; AAT 9806107*  
 [Abstract](#)  [24 Page Preview](#)  [Full Text - PDF](#)  [Order a copy](#)
- 


1-5 of 5

Want to be notified of new results for this search? [Set Up Alert](#) 

Results per

Did you find what you're looking for? If not, [refine your search](#) below or try these suggestions.**Suggested Topics** [About](#)< Previous | [Next >](#)**Browse Suggested Publications** [About](#) < Pre[Cache](#)[Cache AND Computer science](#)[Cache AND Internet](#)[Cache AND Algorithms](#)[M2 Presswire; Coventry](#)[The New Yorker; New York](#)[IEEE Micro; Los Alamitos](#)

10/7/26, 295

**Basic Search**[Tools:](#) [Search Tips](#) [Browse Topics](#) [1 Recent Searches](#)**Search****Clear**Database:   [Select multiple databases](#)Date range:  Limit results to: ☐ Full text documents only ☐ Scholarly journals, including peer-reviewed  [About](#)[More Search Options](#)

---

Copyright © 2007 ProQuest Information and Learning Company. All rights reserved. [Terms and Conditions](#)

[Text-only interface](#)